Machine Learning Assignment - 1

Group Members - Vinayak Singla(B15CS039), Shreyas Malakarjun Patil(B15EE035)

A. Linear Regression

For linear regression we find coeff of w using

```
coeff = np.matmul(pinv(A),B)

Where A and B can be calculated as -
for i in range(n):
    A = A + np.matmul(X[i].reshape(m,1),np.transpose(X[i].reshape(m,1)))
    B = B + y[i] * X[i].reshape(m,1)
```

Result - We achieved a loss of .028 on diabetes dataset from sklearn.

Usage :

```
>>python LinearRegression.py
```

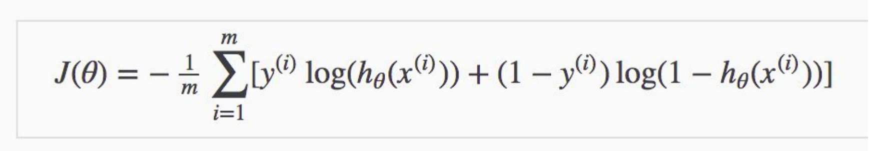
B. Polynomial Regression

It is similar to linear regression only change is the input vector. Let x id the input feature then from it we can for any polynomial p(x) we can form input vector corresponding to p(x).

Eg. let's say p is $x^3 + x^2 + x + 1$ then input vector will become $<x^3 , x^2 , x , 1>$.

C. Logistic Regression

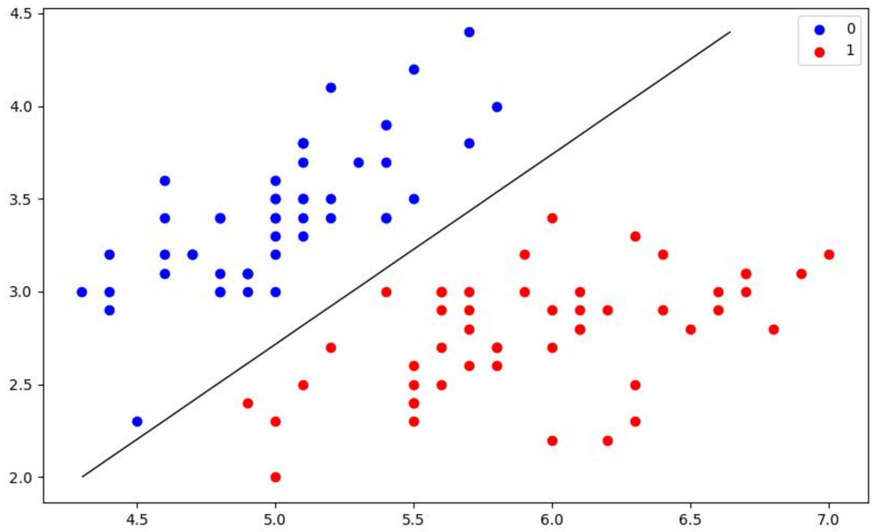
Logistic loss is given as -



$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

As this function is convex we found out the w giving minimum loss by gradient descent. We used dataset iris from sklearn.

Result -



Scatter plot of 2 classes and solution w shown as black line

Usage :

```
>>python LogisticRegression.py
```

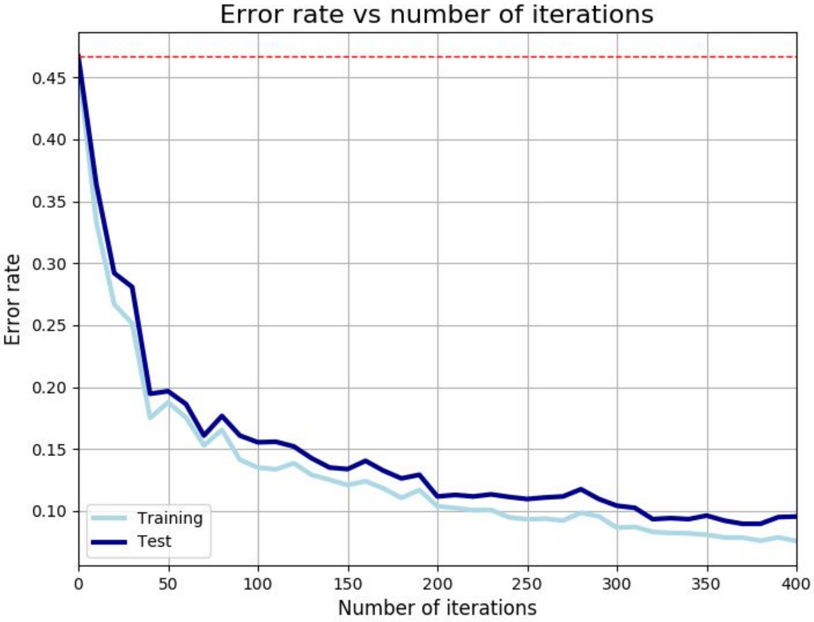
D. Adaboost

For Adaboost we use a inbuilt weak learner - Decision Tree with depth 1 which is same as stump algorithm discussed in class. We used dataset make_hastie_10_2 from sklearn.

Usage :

```
>>python Adaboost.py
```

Result -



Error Rate vs Number of Iterations